Decision Trees

Introduction

What is Segmentation?

What is Segmentation?

- Imagine a scenario where we want to run a SMS marketing campaign to attract more customers in the next quarter
  - Some customers like to see high discount
  - Some customers want to see a large collection of items
  - Some customers are fans of particular brands
  - Some customers are Male some are Female
- Divide them based on their demographics, buying patterns and profile related attributes

What is Segmentation?

- One size doesn't fit all
- Divide the population in such a way that
  - Customers inside a group are homogeneous
  - Customers across groups are heterogeneous
- Is there any statistical way of dividing them correctly based on the data

Segmentation Business Problem

The Business Problem

Old Data

Gender	Marital Status	Ordered the product
M	Married	No
F	Unmarried	Yes
M	Married	No
M	Married	No
M	Married	No
M	Married	No
F	Unmarried	Yes
M	Unmarried	Yes
F	Married	No
M	Married	No
F	Married	No
M	Unmarried	No
F	Married	No
F	Unmarried	Yes

New Data

Gender	Marital Status	Product order
M	Married	??
F	Unmarried	??

The Business Problem

Old Data

Sr No	Gender	Marital Status	Ordered the product
1	M	Married	No
2	F	Unmarried	Yes
3	M	Married	No
4	M	Married	No
5	M	Married	No
6	M	Married	No
7	F	Unmarried	Yes
8	M	Unmarried	Yes
9	F	Married	No
10	M	Married	No
11	F	Married	No
12	M	Unmarried	No
13	F	Married	No
14	F	Unmarried	Yes

New Data

Gender	Marital Status	Product order
M	Married	??
F	Unmarried	??

The Decision Tree Philosophy

The Data

Old Data

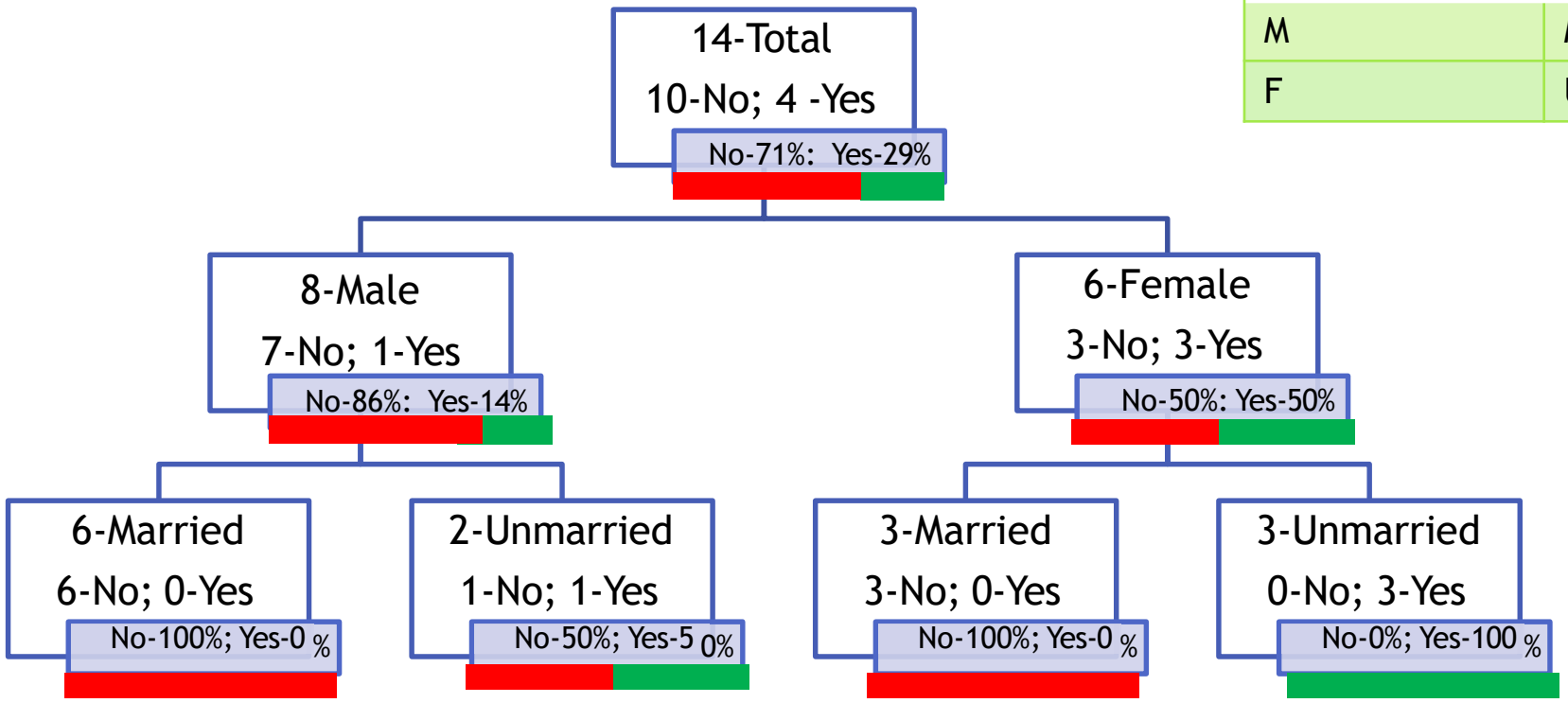
Sr No	Gender	Marital Status	Ordered the product
1	M	Married	No
2	F	Unmarried	Yes
3	M	Married	No
4	M	Married	No
5	M	Married	No
6	M	Married	No
7	F	Unmarried	Yes
8	M	Unmarried	Yes
9	F	Married	No
10	M	Married	No
11	F	Married	No
12	M	Unmarried	No
13	F	Married	No
14	F	Unmarried	Yes

New Data

Gender	Marital Status	Product order
M	Married	??
F	Unmarried	??

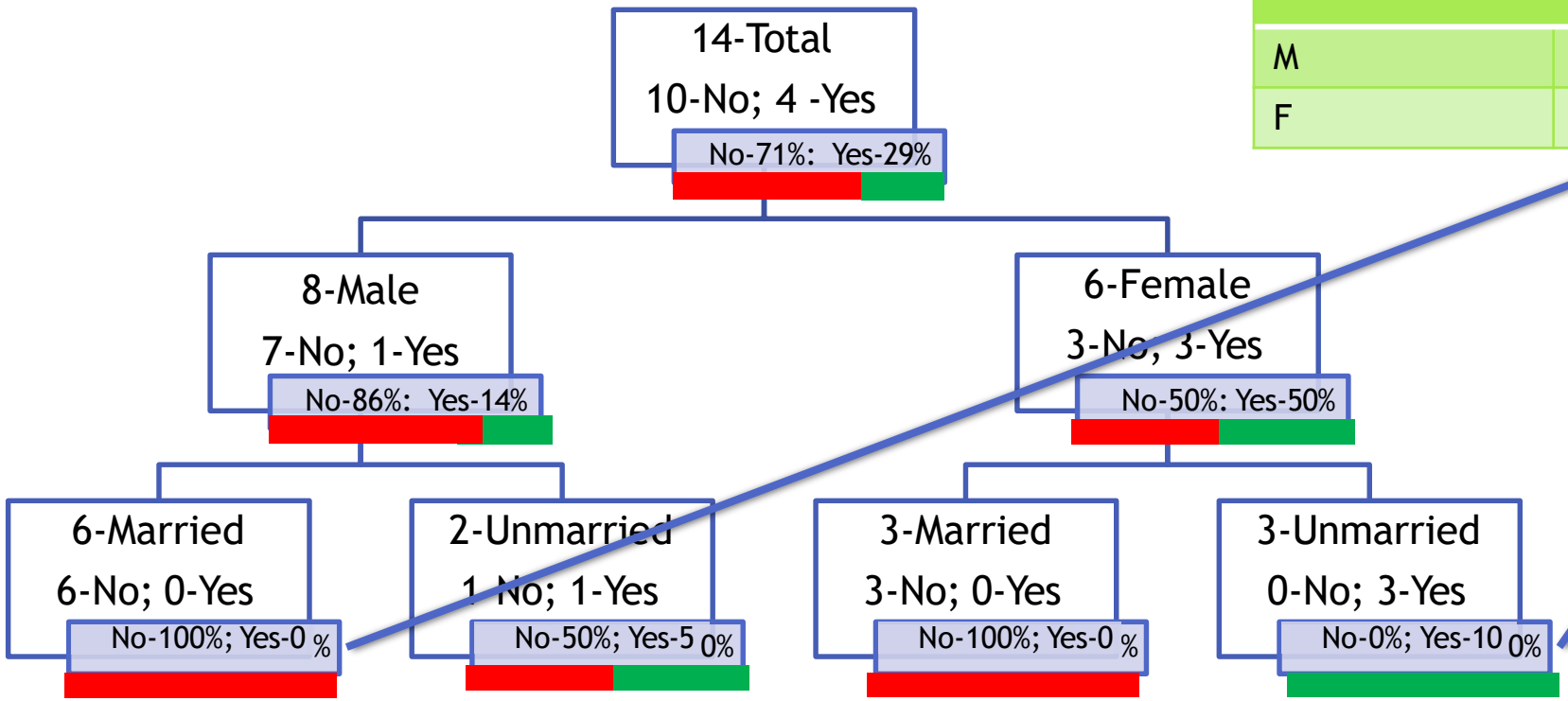
Re-Arranging the data

New Data		
Gender	**Marital Status**	**Product order**
M	Married	??
F	Unmarried	??



Re-Arranging the data

New Data		
Gender	Marital Status	Product order
M	Married	??
F	Unmarried	??

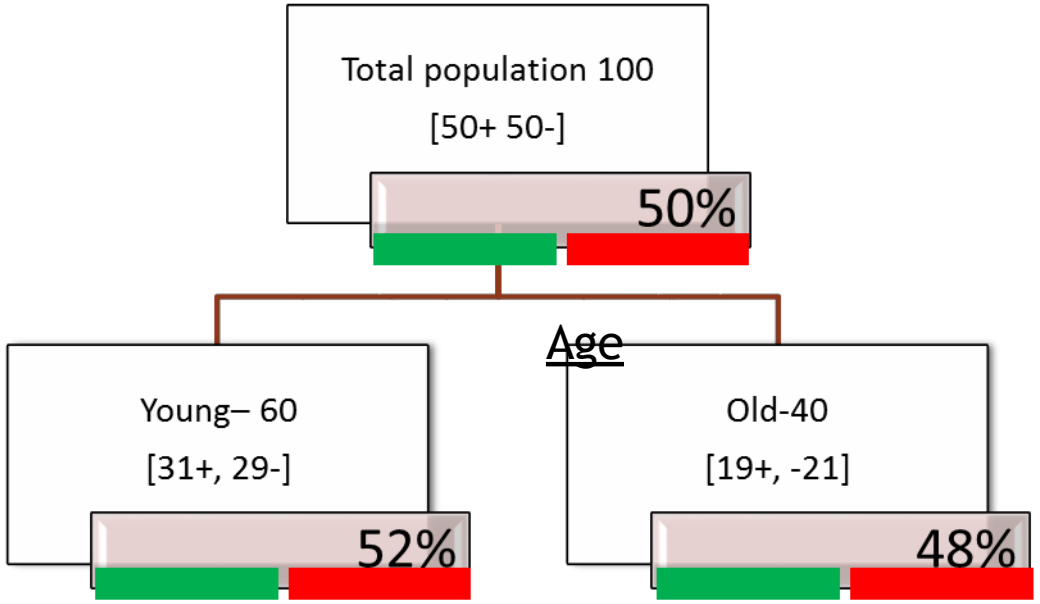


The Decision Tree Approach

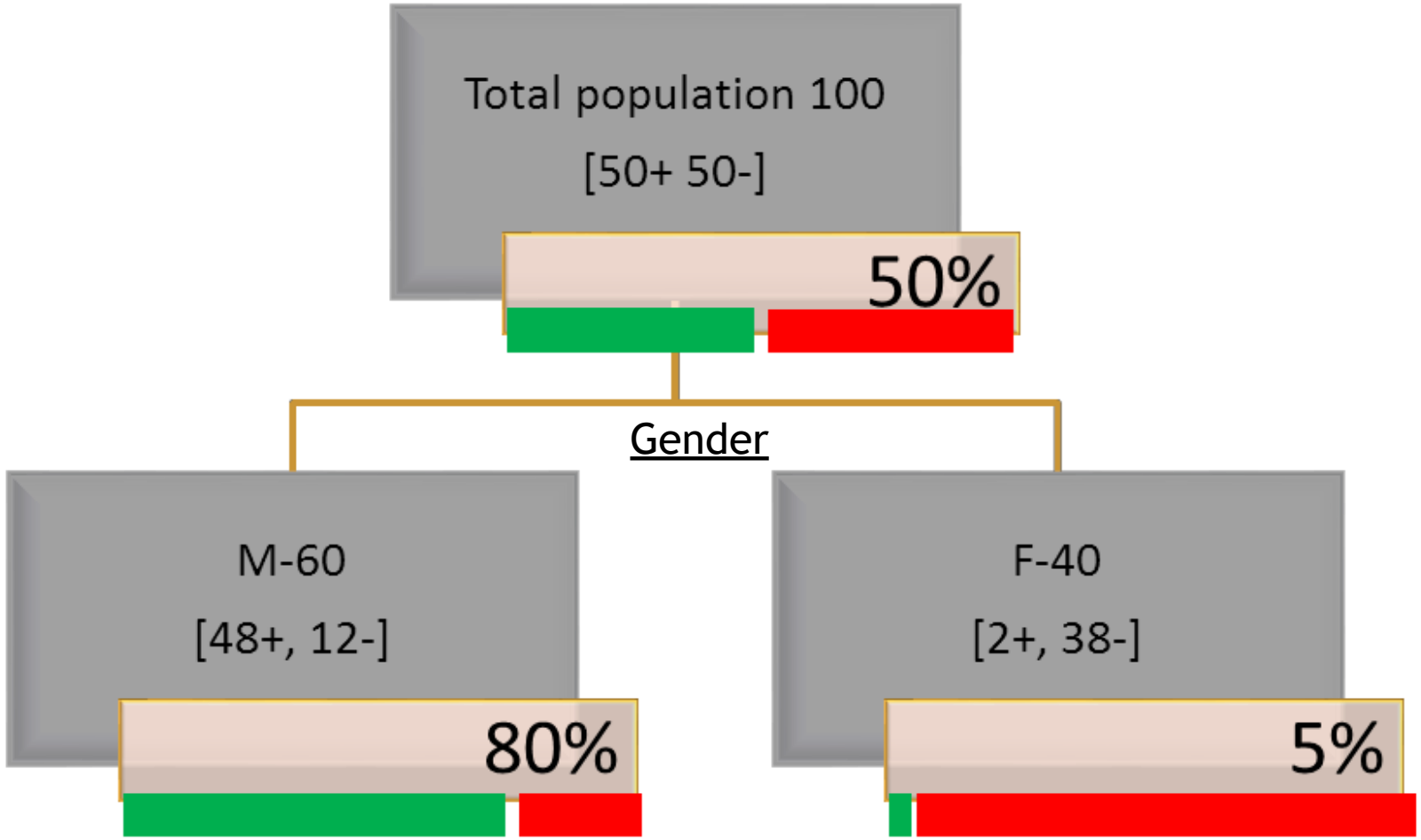
The Decision Tree Approach

- The aim is to decide the whole population or the data set into segments
- The segmentation need to be useful for business decision making.
- If one class is really dominating in a segments
  - Then it will be easy for us to classify the unknown items
  - Then its very easy for applying business strategy
- For example:
  - It takes no great skill to say that the customers have 50% chance to buy and 50% chance to not buy.
  - A good splitting criterion segments the customers with 90% -10% buying probability, say Gender="Female" customers have 5% buying probability and 95% not buying

Example Sales Segmentation Based on Age



Example Sales Segmentation Based on Gender



Main questions

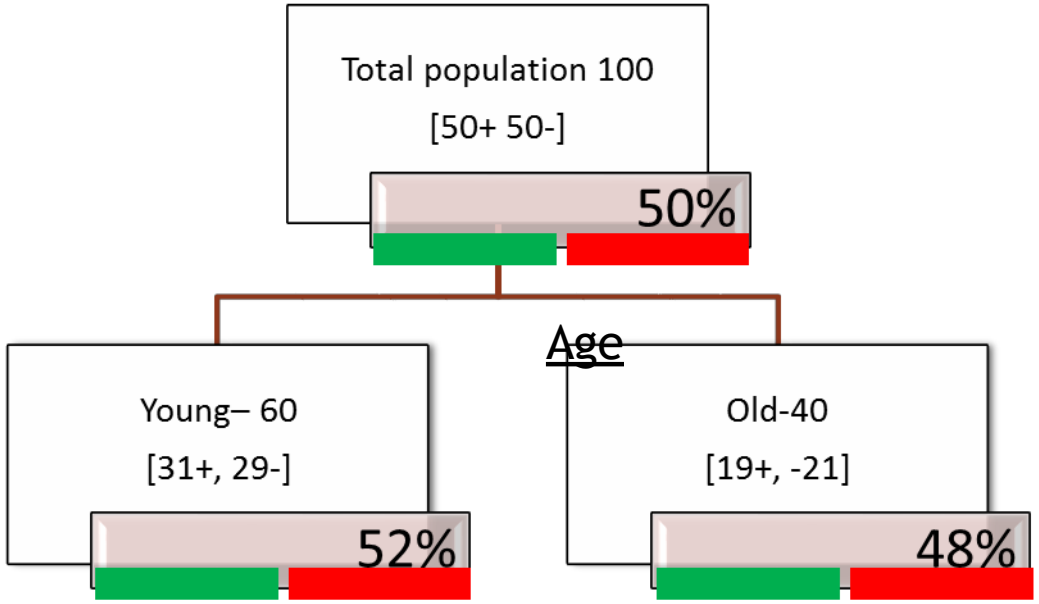
- Ok we are looking for pure segments
- Dataset has many attributes
- Which is the right attribute for pure segmentation?
- Can we start with any attribute?
- Which attribute to start? – The best separating attribute
- Customer Age can impact the sales, gender can impact sales , customer place and demographics can impact the sales. How to identify the best attribute and the split?

The Splitting Criterion

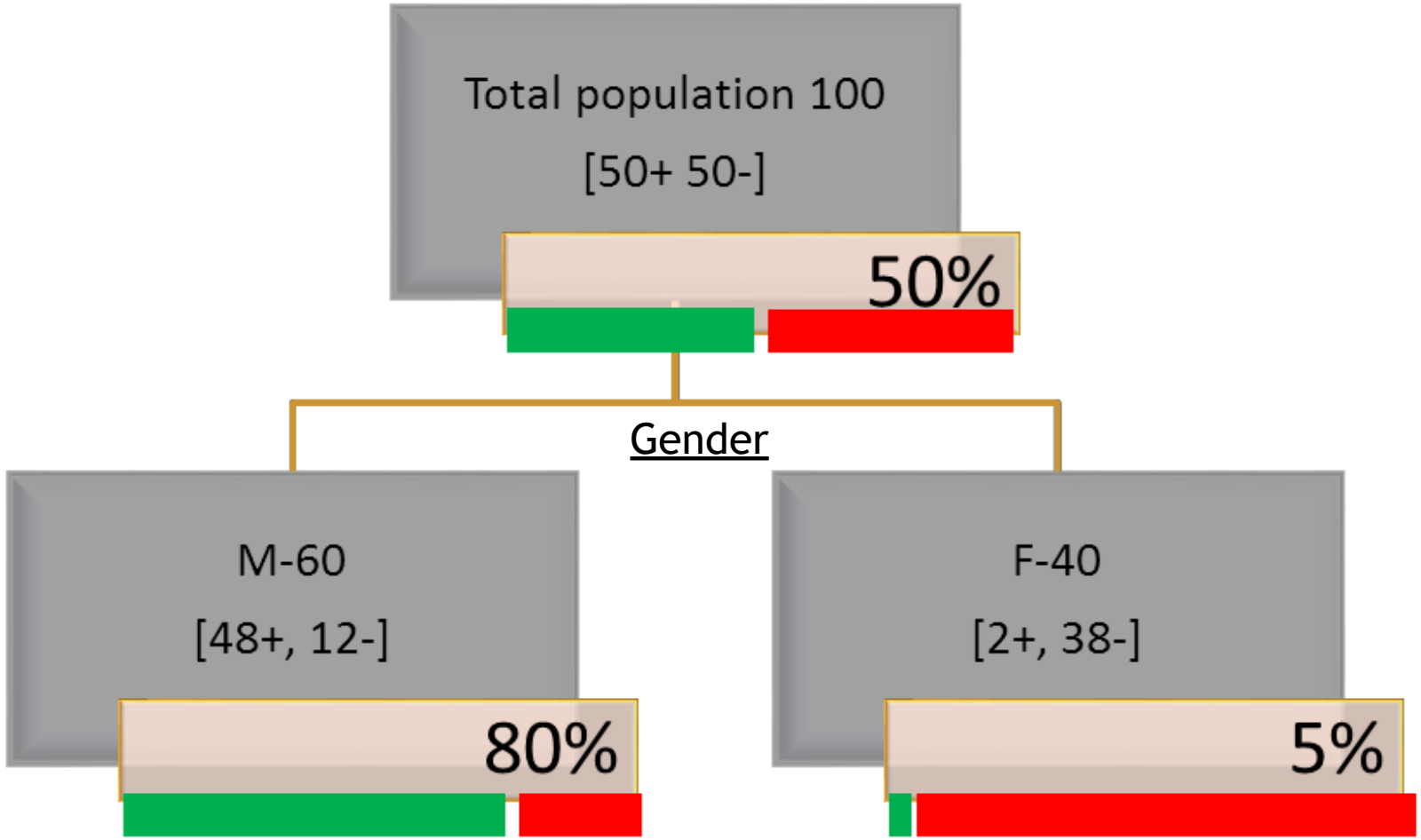
The Splitting Criterion

- The best split is
  - The split does the best job of separating the data into groups
    - Where a single class(either 0 or 1) predominates in each group

Example Sales Segmentation Based on Age

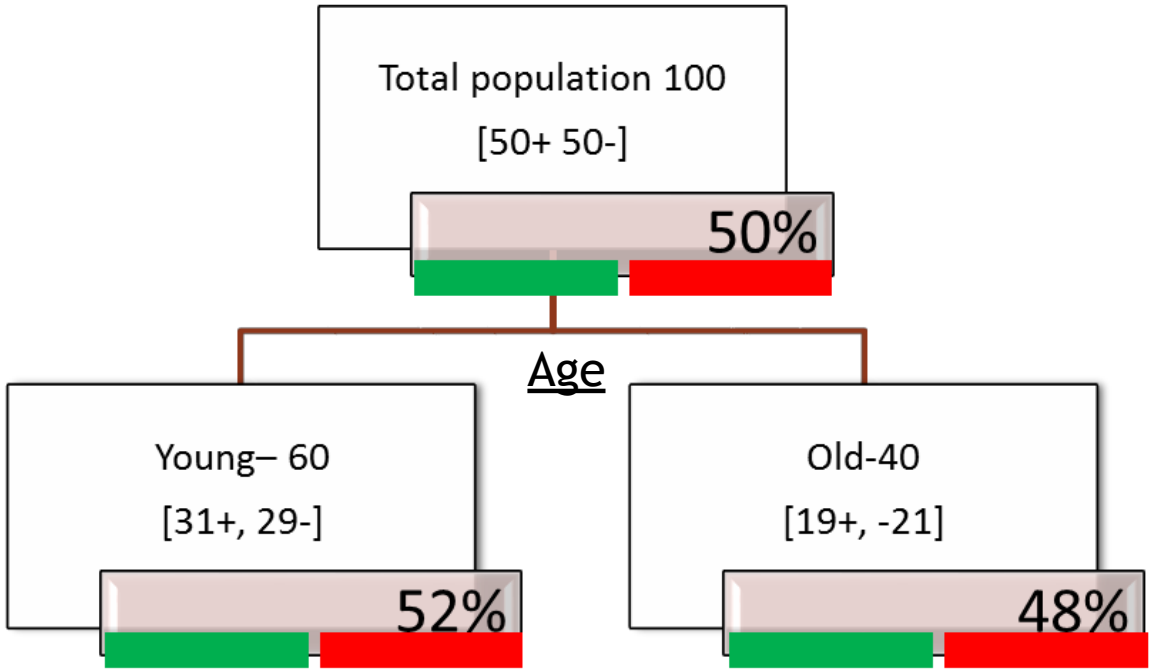


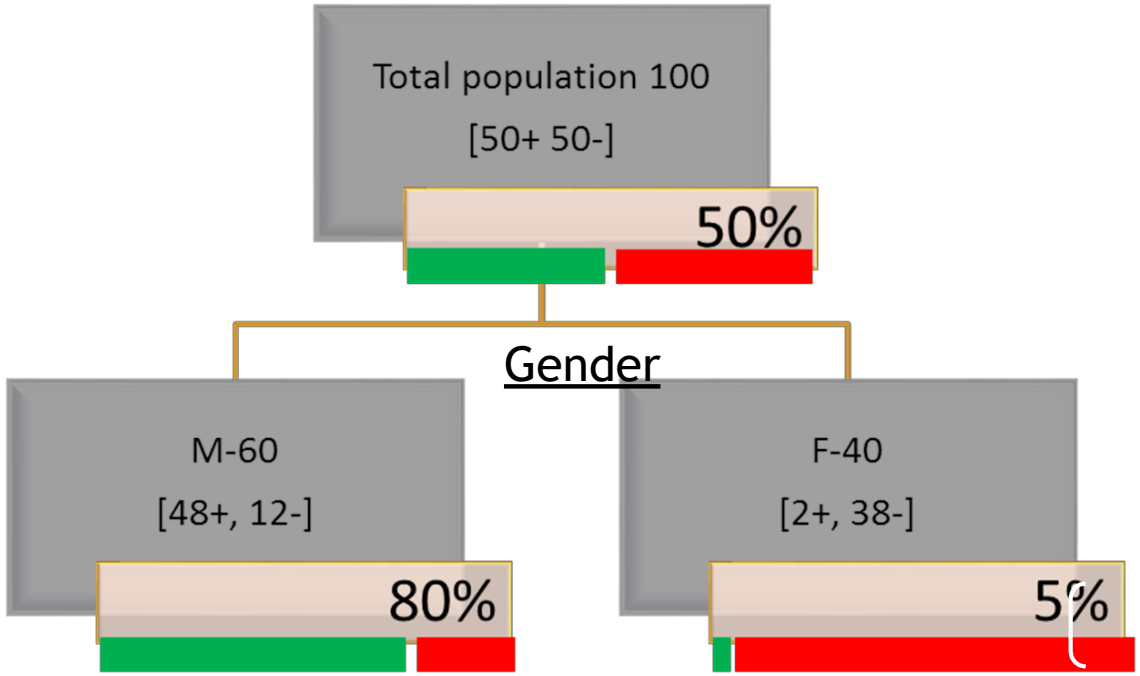
Example Sales Segmentation Based on Gender



Impurity (Diversity) Measures:

- We are looking for a impurity or diversity measure that will give high score for this Age variable(high impurity while segmenting), Low score for Gender variable(Low impurity while segmenting)



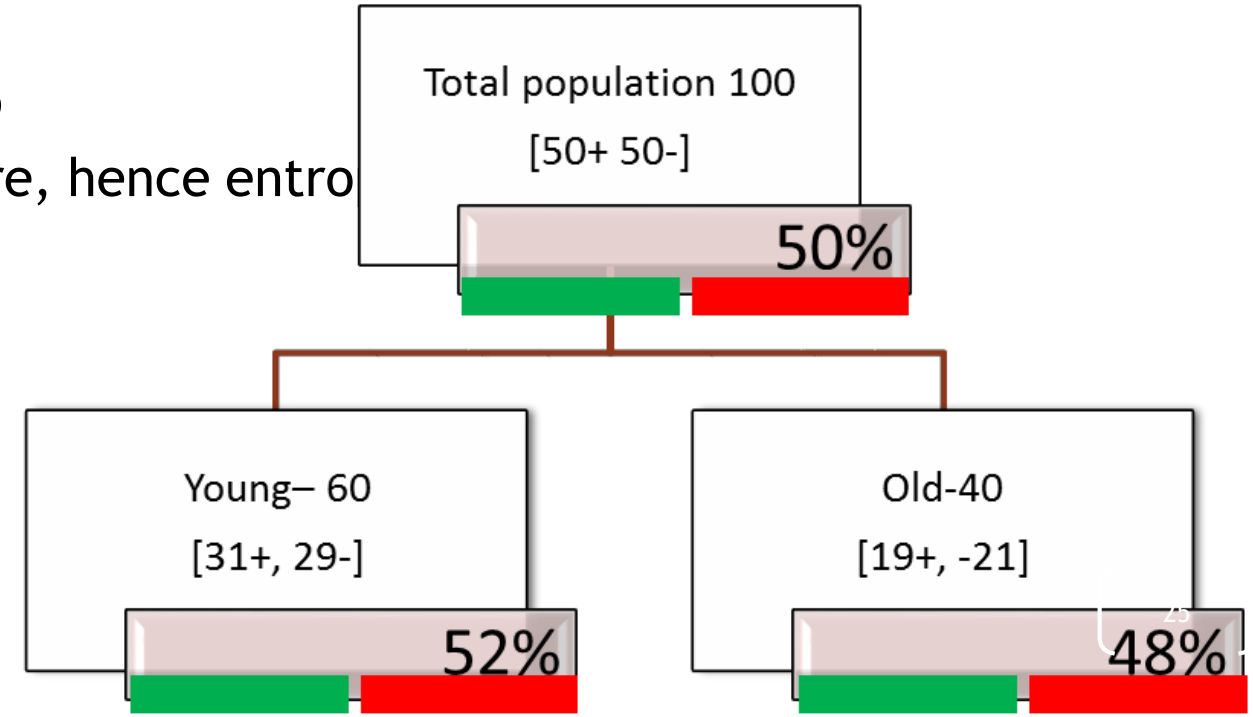


Impurity (Diversity) Measures:

- **Entropy:** Characterizes the impurity/diversity of segment
- Measure of uncertainty/Impurity
- Entropy measures the information amount in a message
- S is a segment of training examples, p_+ is the proportion of positive examples, p_- is the proportion of negative examples
- $\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$
  - Where p_+ is the probabailty of positive class and p_- is the probabailty of negative class
- Entropy is highest when the split has p of 0.5.
- Entropy is least when the split is pure .ie p of 1

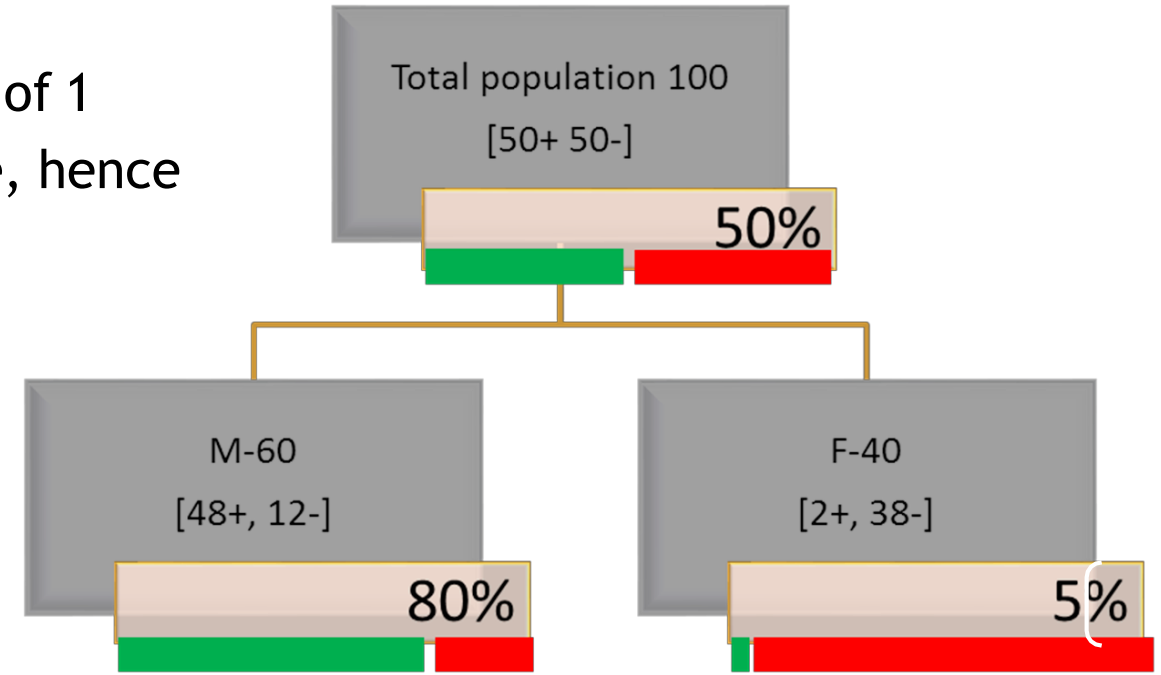
Entropy is highest when the split has p of 0.5

- S is a segment of training examples, p_+ is the proportion of positive examples, p_- is the proportion of negative examples
- Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
- Entropy is highest when the split has p of 0.5
- 50-50 class ratio in a segment is really impure, hence entro
  - Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
  - Entropy(S) = $-0.5*\log_2(0.5) - 0.5*\log_2(0.5)$
  - Entropy(S) = 1



Entropy is least when the split is pure .ie p of 1

- S is a segment of training examples, p_+ is the proportion of positive examples, p_- is the proportion of negative examples
- Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
  - Entropy is least when the split is pure .ie p of 1
  - 100-0 class ratio in a segment is really pure, hence entropy is low
    - Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
    - Entropy(S) = $-1*\log_2(1) - 0*\log_2(0)$
    - Entropy(S) = 0



The less the entropy, the better the split

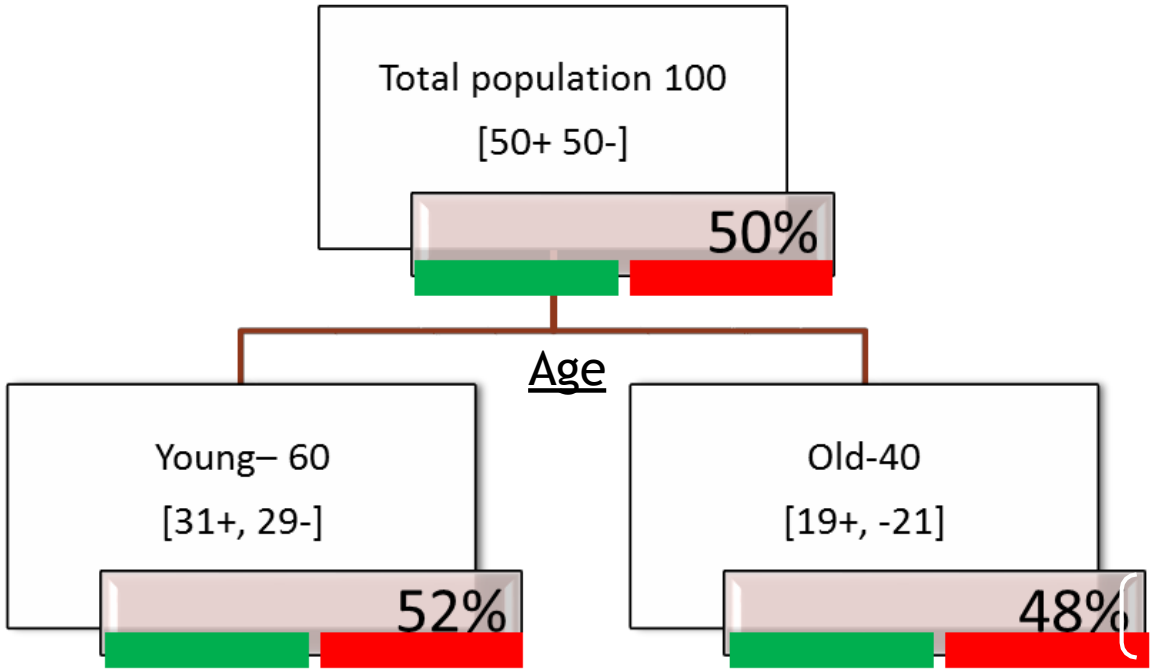
- The less the entropy, the better the split
- Entropy is formulated in such a way that, its value will be high for impure segments

Entropy Calculation -Example

Entropy Calculation

- Entropy at root
- Total population at root 100 [50+,50-]
- Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
- $-0.5 \log_2 (0.5) - 0.5 \log_2 (0.5)$
- $-(0.5)*(-1) - (0.5)*(-1)$
- 1
- 100% Impurity at root

Entropy(S) = -p+ log2 p+ - p- log2 p-

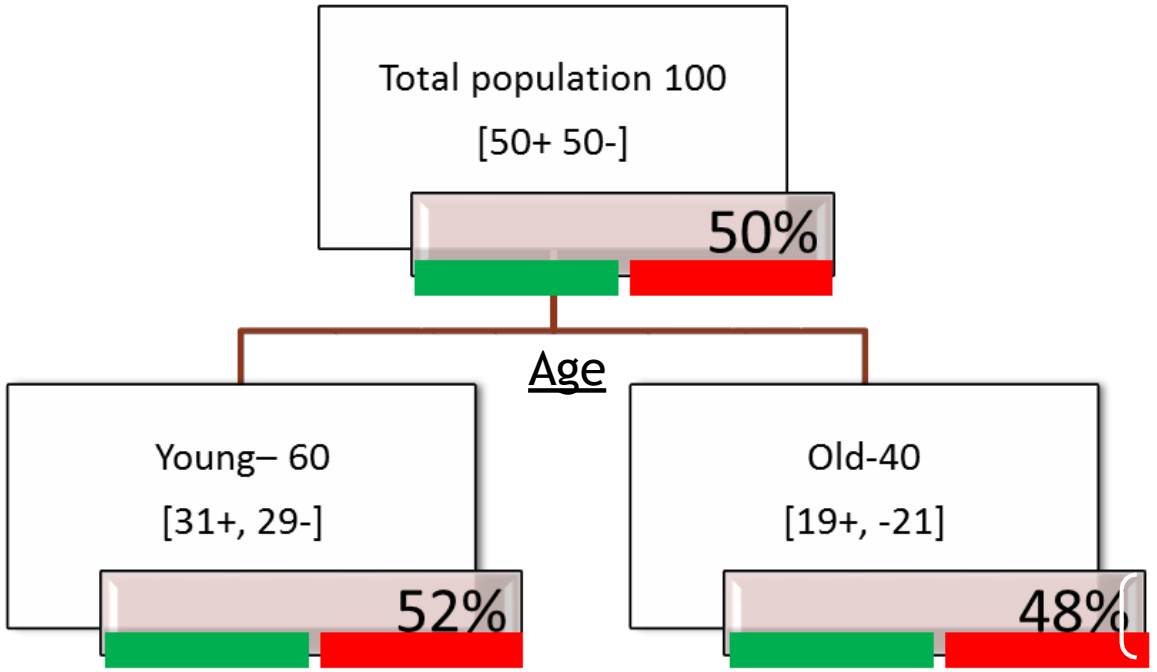


Entropy Calculation

- Gender Splits the population into two segments
- Segment-1 : Age=”Young”
- Segment-2: Age=”Old”

- Entropy at segment-1
- Age=”Young” segment has 60 records [31+,29-]
- Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
- $-31/60 \log_2 31/60 - 29/60 \log_2 29/60$
- (-31/60)*log(31/60,2)-(29/60)*log(29/60,2)
- 0.9991984 (99% Impurity in this segment)

Entropy(S) = -p+ log2 p+ - p- log2 p-

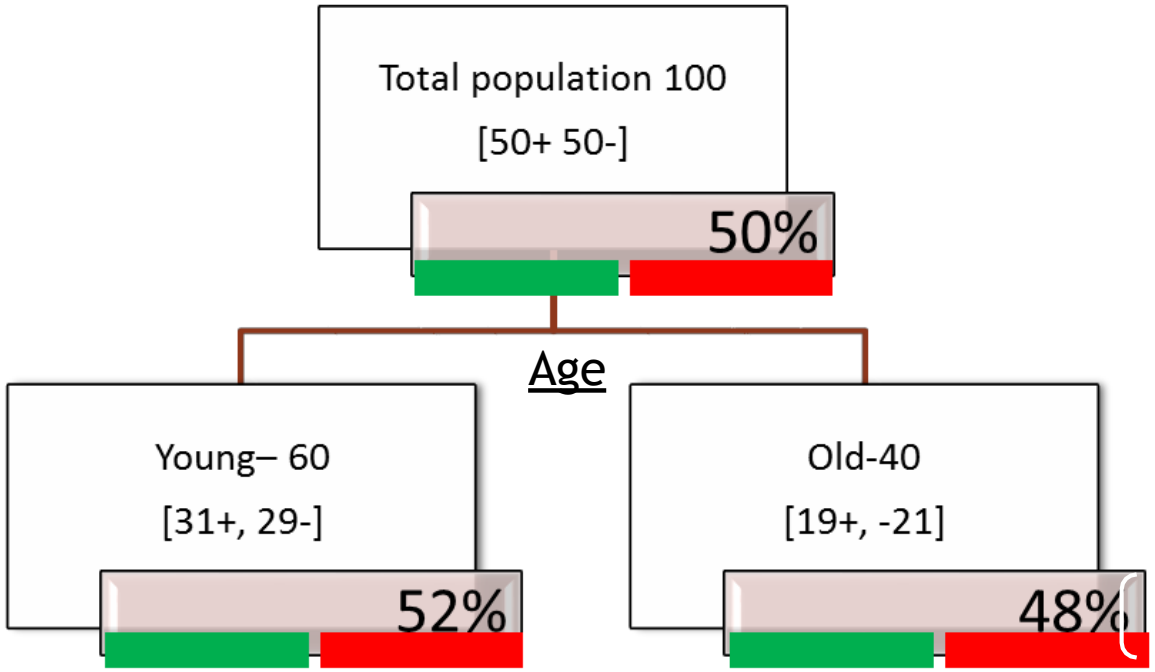


Entropy Calculation

- Gender Splits the population into two segments
- Segment-1 : Age=”Young”
- Segment-2: Age=”Old”

- Entropy at segment-2
- Age=”Old” segment has 40 records [19+,21-]
- Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$
- $-19/40 \log_2 19/40 - 21/40 \log_2 21/40$
- (-19/40)*log(19/40,2)-(21/40)*log(21/40,2)
- 0.9981959(99% Impurity in this segment too)

Entropy(S) = -p+ log2 p+ - p- log2 p-



LAB: Decision Tree Building

LAB: Decision Tree Building

- Data:Ecom_Cust_Relationship_Management/Ecom_Cust_Survey.csv
- How many customers have participated in the survey?
- Overall most of the customers are satisfied or dis-satisfied?
- Can you segment the data and find the concentrated satisfied and dis-satisfied customer segments ?
- What are the major characteristics of satisfied customers?
- What are the major characteristics of dis-satisfied customers?

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - D:\Dropbox\0. Video Proj Files\Python ML\Session 3. Decision Tree\Active Presenter files\3.Decision_Trees_V3.py

3.Decision_Trees_V3.py*

```
#We will need to install graphviz tool in our system and set the path in environment variables.
#Visit http://www.graphviz.org/Download..php and find the optimal version for the computer.
#Get the path for gvedit.exe in install directory(for me it was "C:\Program Files (x86)\Graphviz2.38\
#goto start->computer->system properties->advanced settings->environment variables and add the path.
#We will need python package pydotplus(for older python versions pydot)
#use this command in your anaconda prompt: conda install -c conda-forge pydotplus
#if an error regarding version occure while installing the package go to https://anaconda.org/search/
#this link will show the channel name of the suitable version suitable.
#and we can use use : conda install -c <channel name here> pydotplus

import pydotplus
dot_data = StringIO()
tree.export_graphviz(clf,
                     out_file = dot_data,
                     feature_names = features,
                     filled=True, rounded=True,
                     impurity=False)

graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())

#by looking at the plot we can answer two questions:
#Q.5 What are the major characteristics of satisfied customers?

#Major_characteristics= Order.Quantity<40 & Age<30 | Order.Quantity >=40

#Q 6. What are the major characteristics of dis-satisfied customers?
#Major_characteristics= Order.Quantity<40 & Age>=30

#LAB : Tree Validation
########################################
##########Tree Validation
#Tree Validation
predict1 = clf.predict(X)
predict1

from sklearn.metrics import confusion_matrix ###for using confusion matrix###
cm = confusion_matrix(y, predict1)
print (cm)
```

Variable explorer

Name	Type	Size	Value
X	DataFrame	(11805, 4)	Column names: Age, Order_Quantity, Customer_Type, Improvement_Area
df	DataFrame	(11805, 7)	Column names: Cust_num, Region, Age, Order_Quantity, Customer_Type, Improvement_Area, Overall_Satisfaction
features	list	4	['Age', 'Order_Quantity', 'Customer_Type', 'Improvement_Area']
satisfied	int64	1	class 'numpy.int64'
y	Series	(11805,)	class 'pandas.core.series.Series'

Variable explorer | File explorer | Help

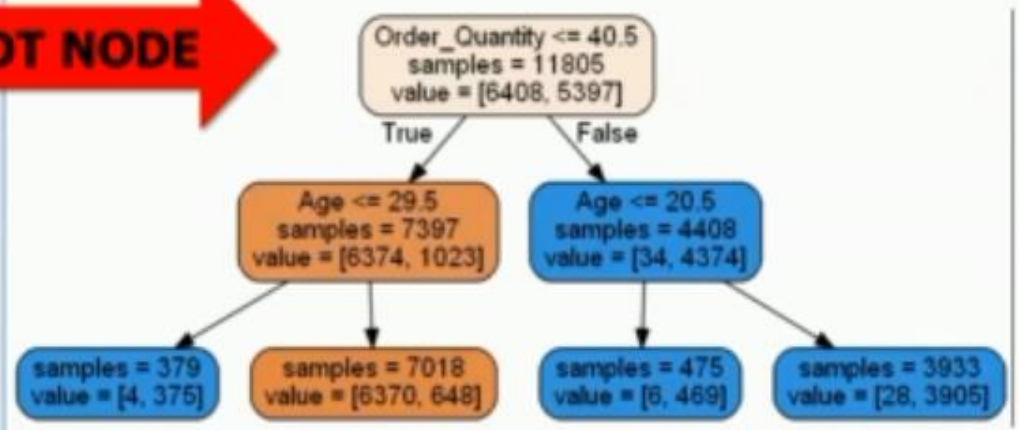
IPython console

Console 1/A

```
...:
...: graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
...: Image(graph.create_png())
...:
```

Out[44]:

ROOT NODE



In [45]:

Type here to search

1:47 AM
7/19/2019

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\StatInfer

Editor - D:\Dropbox\0. Video Proj Files\Python ML\Session 3. Decision Tree\Active Presenter files\3.Decision_Trees_V3.py

3.Decision_Trees_V3.py*

```
#Import Data
import pandas as pd

##Ecom_Cust_Survey = pd.read_csv('...',header = 0)
df = pd.read_csv('D:\\Google Drive\\Training\\Datasets\\Ecom_Cust_Relationship_Management\\Ecom_Cust_S

df.dropna(inplace='True') # to remove all the missing values rows..
#Q 1. How many customers have participated in the survey?
df.info()
#ANS: 11805

#Q.2 Overall most of the customers are satisfied or dis-satisfied?
#total number of customers
df.shape
df.head()


#number of satisfied customers
satisfied = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).astype(int).sum()
satisfied
#number of dis-satisfied customers
df.shape[0]-satisfied


#Q 3. Can you segment the data and find the concentrated satisfied and dis-satisfied customer segments
#solution:
# We will create a tree model in python using the sci-kit module
# before that we will need to convert most of the feature data into numerical or hash values as scikit
# Welcome to variable transformation

df['Region'] = df['Region'].map( {'EAST': 1, 'WEST': 2, 'NORTH': 3, 'SOUTH':4} ).astype(int)
df['Customer_Type'] = df['Customer_Type'].map({'Prime': 1, 'Non_Prime': 0}).astype(int)

#We will also need to change the column names, as '.' and spaces are part of many basic funcions in py
df.rename(columns={'Order Quantity':'Order_Quantity', 'Improvement Area' :'Improvement_Area'}, inplace
df['Improvement_Area'] = df['Improvement_Area'].map({'Website UI':1, 'Packing & Shipping':2, 'Product
df['Overall_Satisfaction'] = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).as

#Need the library to create the tree
from sklearn.tree import DecisionTreeClassifier

#Defining Features and Lables
```

Variable explorer

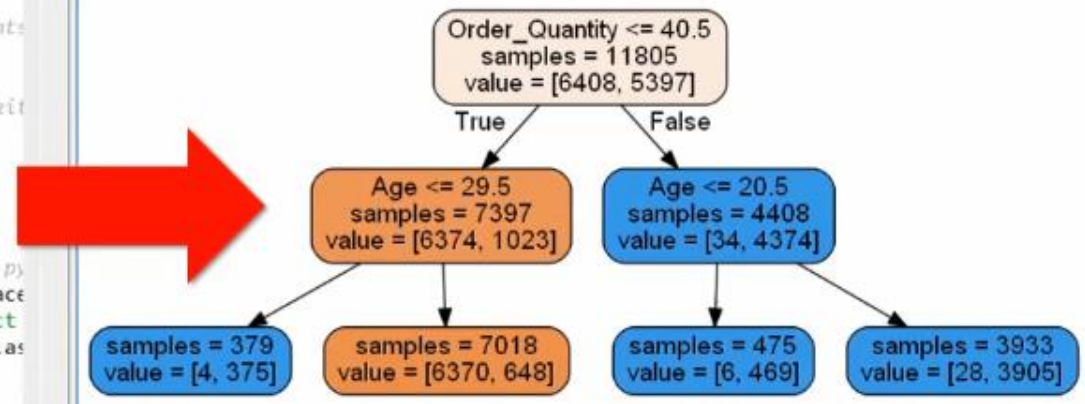
Name	Type	Size	Value
X	DataFrame	(11805, 4)	Column names: Age, Order_Quantity, Customer_Type, Improvement_Area
df	DataFrame	(11805, 7)	Column names: Cust_num, Region, Age, Order_Quantity, Customer_Type, Improvement_Area, Overall_Satisfaction
features	list	4	['Age', 'Order_Quantity', 'Customer_Type', 'Improvement_Area']
satisfied	int64	1	class 'numpy.int64'
y	Series	(11805,)	class 'pandas.core.series.Series'

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
    ...:
    ...: graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
    ...: Image(graph.create_png())
    ...:
Out[44]:
```




```
In [45]:
```

Python console | History log | IPython console

Type here to search

1:48 AM
7/19/2019

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\StatInfer

Editor - D:\Dropbox\0. Video Proj Files\Python ML\Session 3. Decision Tree\Active Presenter files\3.Decision_Trees_V3.py

3.Decision_Trees_V3.py*

```
#Import Data
import pandas as pd

##Ecom_Cust_Survey = pd.read_csv('...',header = 0)
df = pd.read_csv('D:\\Google Drive\\Training\\Datasets\\Ecom_Cust_Relationship_Management\\Ecom_Cust_S

df.dropna(inplace='True') # to remove all the missing values rows..
#Q 1. How many customers have participated in the survey?
df.info()
#ANS: 11805

#Q.2 Overall most of the customers are satisfied or dis-satisfied?
#total number of customers
df.shape
df.head()


#number of satisfied customers
satisfied = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).astype(int).sum()
satisfied
#number of dis-satisfied customers
df.shape[0]-satisfied


#Q 3. Can you segment the data and find the concentrated satisfied and dis-satisfied customer segments
#solution:
# We will create a tree model in python using the sci-kit module
# before that we will need to convert most of the feature data into numerical or hash values as scikit
# Welcome to variable transformation

df['Region'] = df['Region'].map( {'EAST': 1, 'WEST': 2, 'NORTH': 3, 'SOUTH':4} ).astype(int)
df['Customer_Type'] = df['Customer_Type'].map({'Prime': 1, 'Non_Prime': 0}).astype(int)

#We will also need to change the column names, as '.' and spaces are part of many basic fu
df.rename(columns={'Order Quantity':'Order_Quantity'
df['Improvement_Area'] = df['Improvement_Area'].map(
df['Overall_Satisfaction'] = df['Overall_Satisfactio

#Need the library to create the tree
from sklearn.tree import DecisionTreeClassifier

#Defining Features and Lables
```

Rule 1. Concentrated Satisfied

Variable explorer

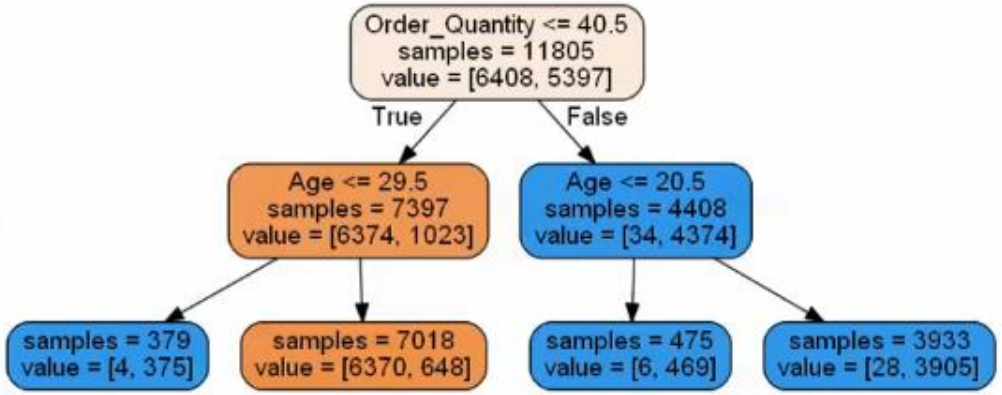
Name	Type	Size	Value
X	DataFrame	(11805, 4)	Column names: Age, Order_Quantity, Customer_Type, Improvement_Area
df	DataFrame	(11805, 7)	Column names: Cust_num, Region, Age, Order_Quantity, Customer_Type, Improvement_Area, Overall_Satisfaction
features	list	4	['Age', 'Order_Quantity', 'Customer_Type', 'Improvement_Area']
satisfied	int64	1	class 'numpy.int64'
y	Series	(11805,)	class 'pandas.core.series.Series'

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
...:
...: graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
...: Image(graph.create_png())
...:
Out[44]:
```




```
In [45]:
```

Python console | History log | IPython console

Type here to search

1:48 AM
7/19/2019

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - D:\Dropbox\0. Video Proj Files\Python ML\Session 3. Decision Tree\Active Presenter files\3.Decision_Trees_V3.py

3.Decision_Trees_V3.py*

```
#Import Data
import pandas as pd

##Ecom_Cust_Survey = pd.read_csv('...',header = 0)
df = pd.read_csv('D:\\Google Drive\\Training\\Datasets\\Ecom_Cust_Relationship_Management\\Ecom_Cust_S

df.dropna(inplace='True') # to remove all the missing values rows..
#Q 1. How many customers have participated in the survey?
df.info()
#ANS: 11805

#Q.2 Overall most of the customers are satisfied or dis-satisfied?
#Total number of customers
df.shape
df.head()


#number of satisfied customers
satisfied = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).astype(int).sum()
satisfied
#number of dis-satisfied customers
df.shape[0]-satisfied


#Q 3. Can you segment the data and find the concentrated satisfied and dis-satisfied customer segments
#solution:
# We will create a tree model in python using the sci-kit module
# before that we will need to convert most of the feature data into numerical or hash values as scikit
# Welcome to variable transformation

df['Region'] = df['Region'].map( {'EAST': 1, 'WEST': 2, 'NORTH': 3, 'SOUTH':4} ).astype(int)
df['Customer_Type'] = df['Customer_Type'].map({'Prime': 1, 'Non_Prime': 0}).astype(int)

#We will also need to change the column names, as '.' and spaces are part of many basic functions in py
df.rename(columns={'Order Quantity':'Order_Quantity', 'Improvement Area' :
df['Improvement_Area'] = df['Improvement_Area'].map({'Website UI':1, 'Pack
df['Overall_Satisfaction'] = df['Overall_Satisfaction'].map( {'Dis Satisfi

#Need the library to create the tree
from sklearn.tree import DecisionTreeClassifier

#Defining features and lables
```

Variable explorer

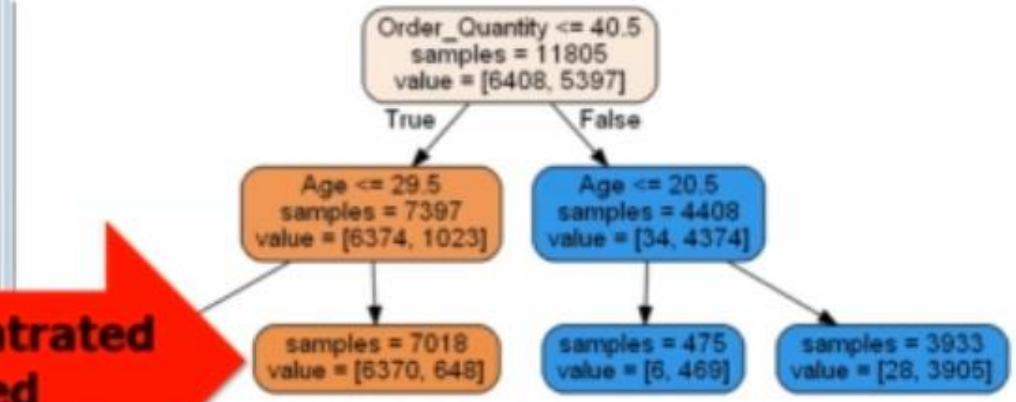
Name	Type	Size	Value
X	DataFrame	(11805, 4)	Column names: Age, Order_Quantity, Customer_Type, Improvement_Area
df	DataFrame	(11805, 7)	Column names: Cust_num, Region, Age, Order_Quantity, Customer_Type, Improvement_Area, Overall_Satisfaction
features	list	4	['Age', 'Order_Quantity', 'Customer_Type', 'Improvement_Area']
satisfied	int64	1	class 'numpy.int64'
y	Series	(11805,)	class 'pandas.core.series.Series'

Variable explorer File explorer Help

IPython console

Console 1/A

```
...:
...: graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
...: Image(graph.create_png())
...:
Out[44]:
```



RULE 2 Consentrated Dissatisfied

```
In [45]:
```

Python console History log IPython console

Type here to search

1:49 AM
7/19/2019

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

```
#Import Data
import pandas as pd

##Ecom_Cust_Survey = pd.read_csv('...',header = 0)
df = pd.read_csv('D:\\Google Drive\\Training\\Datasets\\Ecom_Cust_Relationship_Management\\Ecom_Cust_S

df.dropna(inplace='True') # to remove all the missing values rows..
#Q 1. How many customers have participated in the survey?
df.info()
#ANS: 11805

#Q.2 Overall most of the customers are satisfied or dis-satisfied?
#Total number of customers
df.shape
df.head()


#number of satisfied customers
satisfied = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).astype(int).sum()
satisfied
#number of dis-satisfied customers
df.shape[0]-satisfied


#Q 3. Can you segment the data and find the concentrated satisfied and dis-satisfied customer segments
#solution:
# We will create a tree model in python using the sci-kit module
# before that we will need to convert most of the feature data into numerical or hash values as scikit
# Welcome to variable transformation

df['Region'] = df['Region'].map( {'EAST': 1, 'WEST': 2, 'NORTH': 3, 'SOUTH':4} ).astype(int)
df['Customer_Type'] = df['Customer_Type'].map({'Prime': 1, 'Non_Prime': 0}).astype(int)

#We will also need to change the column names, as '.' and spaces are part of many basic functions in py
df.rename(columns={'Order Quantity':'Order_Quantity', 'Improvement Area' :'Improvement_Area'}, inplace
df['Improvement_Area'] = df['Improvement_Area'].map({'Website UI':1, 'Packing & Shipping':2, 'Product
df['Overall_Satisfaction'] = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).as

#Need the library to create the tree
from sklearn.tree import DecisionTreeClassifier

#Defining features and labels
```

Variable explorer

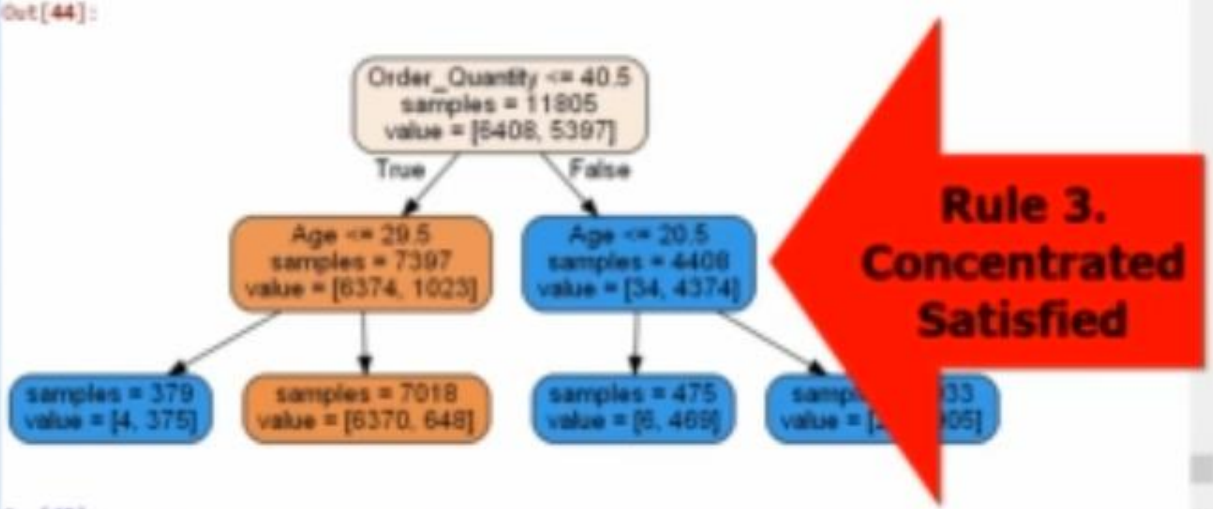
Name	Type	Size	Value
X	DataFrame	(11805, 4)	Column names: Age, Order_Quantity, Customer_Type, Improvement_Area
df	DataFrame	(11805, 7)	Column names: Cust_num, Region, Age, Order_Quantity, Customer_Type, Improvement_Area, Overall_Satisfaction
features	list	4	['Age', 'Order_Quantity', 'Customer_Type', 'Improvement_Area']
satisfied	int64	1	class 'numpy.int64'
y	Series	(11805,)	class 'pandas.core.series.Series'

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
...:
...: graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
...: Image(graph.create_png())
...:
Out[44]:
```




```
In [45]:
```

Type here to search

1:50 AM
7/19/2019

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\StatInfer

Editor - D:\Dropbox\0. Video Proj Files\Python ML\Session 3. Decision Tree\Active Presenter files\3.Decision_Trees_V3.py

3.Decision_Trees_V3.py*

```
satisfied = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).astype(int).sum()
satisfied
#number of dis-satisfied customers
df.shape[0]-satisfied


#Q 3. Can you segment the data and find the concentrated satisfied and dis-satisfied customer segments
#solution:
# We will create a tree model in python using the sci-kit module
# before that we will need to convert most of the feature data into numerical or hash values as scikit
# Welcome to variable transformation

df['Region'] = df['Region'].map( {'EAST': 1, 'WEST': 2, 'NORTH': 3, 'SOUTH':4} ).astype(int)
df['Customer_Type'] = df['Customer_Type'].map({'Prime': 1, 'Non_Prime': 0}).astype(int)

#We will also need to change the column names, as '.' and spaces are part of many basic funcions in py
df.rename(columns={'Order Quantity':'Order_Quantity', 'Improvement Area' :'Improvement_Area'}, inplace
df['Improvement_Area'] = df['Improvement_Area'].map({'Website UI':1, 'Packing & Shipping':2, 'Product
df['Overall_Satisfaction'] = df['Overall_Satisfaction'].map( {'Dis Satisfied': 0, 'Satisfied': 1} ).as

#Need the library to create the tree
from sklearn.tree import DecisionTreeClassifier

#Defining Features and lables
features= list(df.columns[2:6])
features

X = df[features]
y = df['Overall_Satisfaction']

#Building Tree Model
clf = DecisionTreeClassifier(max_depth=2)
clf.fit(X,y)

#What are the major characteristics of satisfied customers?

#Plotting the trees
#Unfortunately drawing a beautiful tree is not easy in python, Still
#you will need to install pydot
#use this command in your anaconda prompt: conda install -c anaconda pydot=1.0.28

from IPython.display import Image
```

Variable explorer

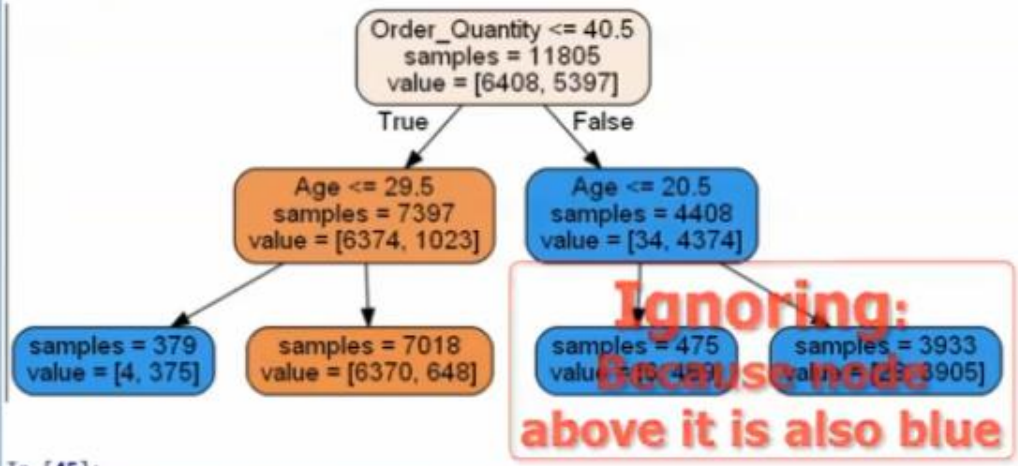
Name	Type	Size	Value
X	DataFrame	(11805, 4)	Column names: Age, Order_Quantity, Customer_Type, Improvement_Area
df	DataFrame	(11805, 7)	Column names: Cust_num, Region, Age, Order_Quantity, Customer_Type, Improvement_Area, Overall_Satisfaction
features	list	4	['Age', 'Order_Quantity', 'Customer_Type', 'Improvement_Area']
satisfied	int64	1	class 'numpy.int64'
y	Series	(11805,)	class 'pandas.core.series.Series'

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
    ...:
    ...: graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
    ...: Image(graph.create_png())
    ...:
Out[44]:
```




```
In [45]:
```

Python console | History log | IPython console

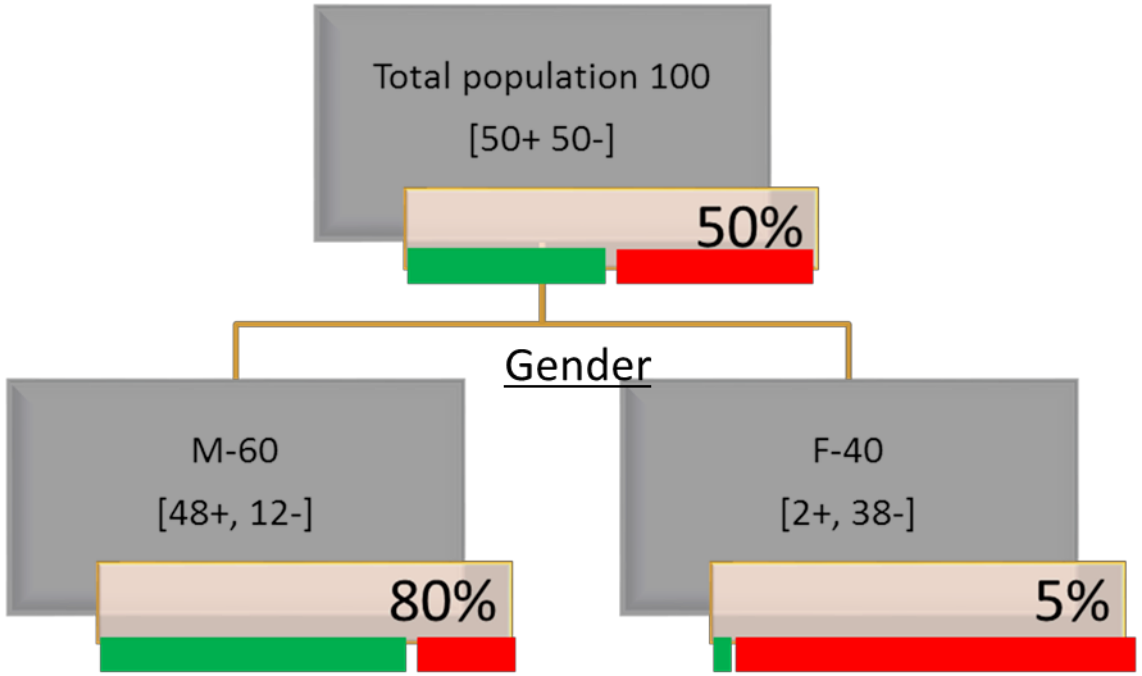
Type here to search

1:51 AM
7/19/2019

LAB: Entropy Calculation – use calculator or Excel

LAB Entropy Calculation

- Calculate entropy at the root for the given population
- Calculate the entropy for the two distinct gender segments



Code- Entropy Calculation

- Entropy at root 100%
- Male Segment : (-48/60)*log(48/60,2)-(12/60)*log(12/60,2)
  - 0.7219281
- FemaleSegment : (-2/40)*log(2/40,2)-(38/40)*log(38/40,2)
  - 0.286397

Thank you